CMSC 28100

Introduction to Complexity Theory

Autumn 2025
Instructor: William Hoza

The complexity class $\mathbf{coNP}$

- Let $Y \subseteq \{0,1\}^*$
- **Definition:** $Y \in \mathrm{coNP}$ if there exists a randomized polynomial-time Turing machine M such that for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ rejects } w] = 0$
  - If $w \notin Y$, then $\Pr[M \text{ rejects } w] \neq 0$

The complexity class $\mathbf{coNP}$

- Let $Y \subseteq \{0,1\}^*$ and let $\bar{Y} = \{0,1\}^* \setminus Y$
- **Fact:** $Y \in \mathrm{NP}$ if and only if $\bar{Y} \in \mathrm{coNP}$
- coNP is the set of complements of languages in NP

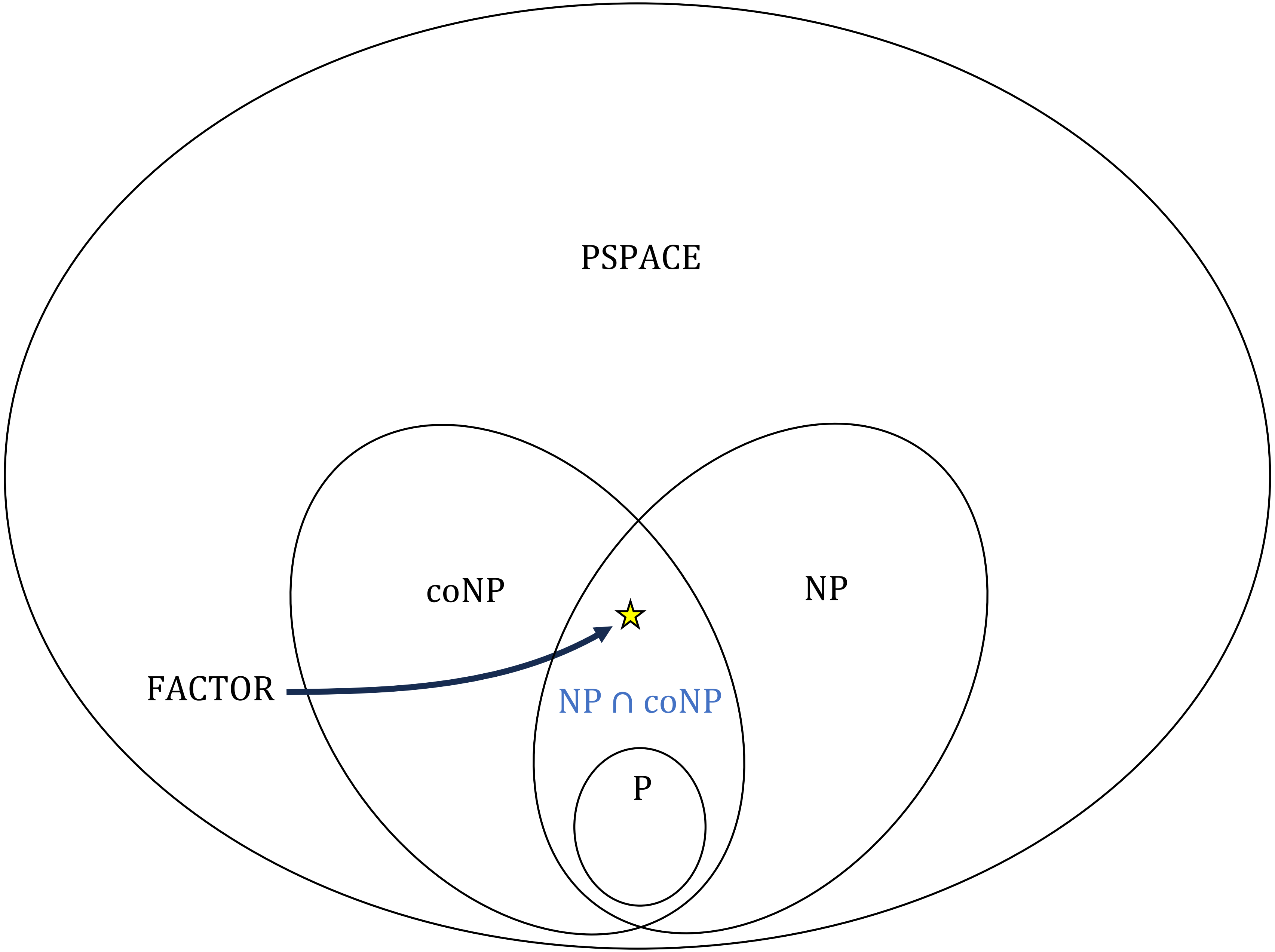
The complexity class $\mathbf{NP} \cap \mathbf{coNP}$

- We have shown that $\text{FACTOR} \in \text{NP}$ and $\text{FACTOR} \in \text{coNP}$
- $\text{FACTOR} \in \text{NP} \cap \text{coNP}$
- $Y \in \text{NP} \cap \text{coNP}$ means that for every instance, there is a certificate
  - A certificate of membership for YES instances
  - A certificate of non-membership for NO instances

The NP vs. coNP problem

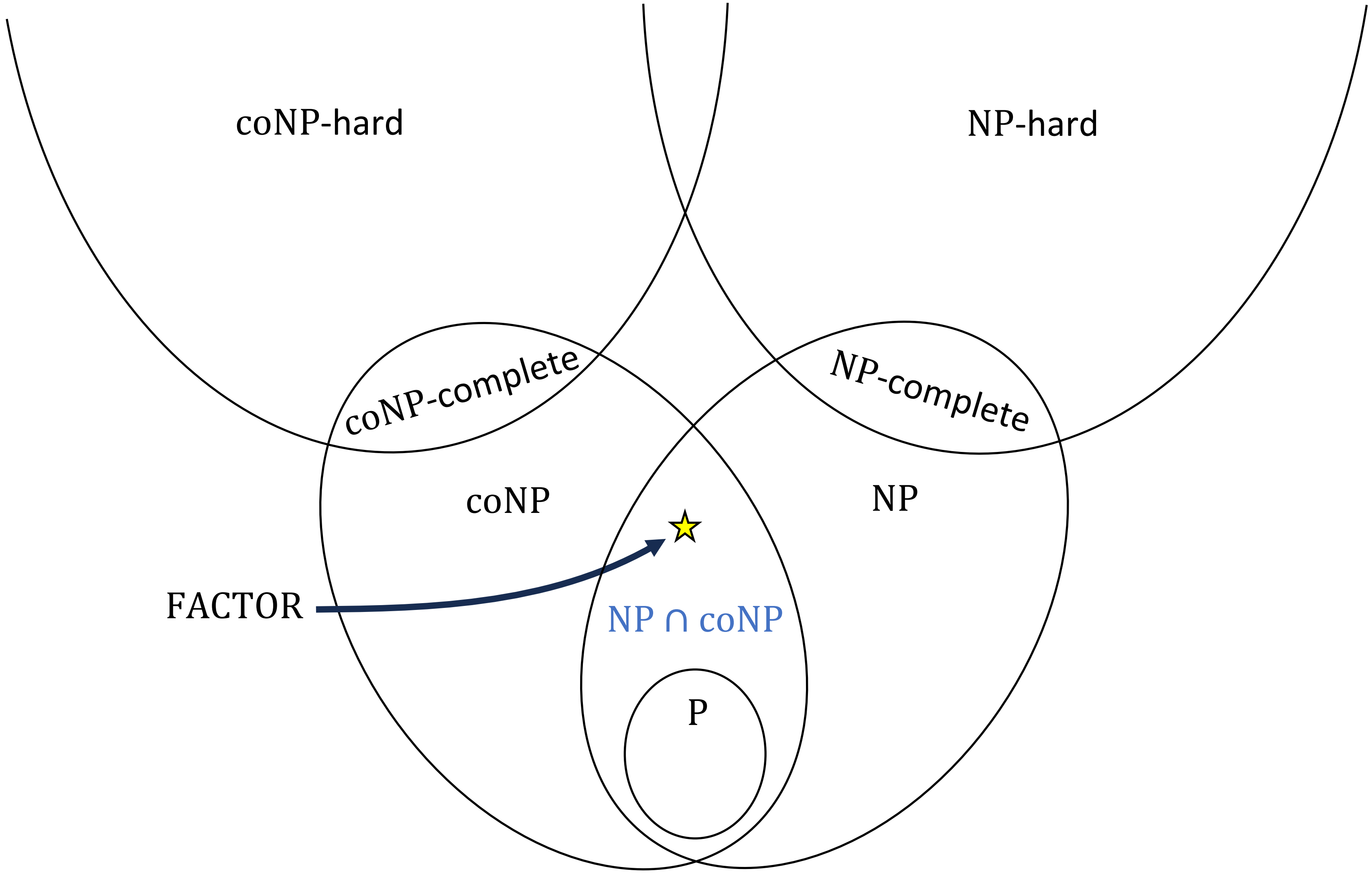
Conjecture: $\mathrm{NP} \neq \mathrm{coNP}$

- "$\mathrm{NP} = \mathrm{coNP}$" would mean that for every unsatisfiable circuit, there is some short certificate I could present to prove to you that a circuit is unsatisfiable
- That sounds counterintuitive! But we don't really know


PSPACE
coNP
NP
FACTOR
NP ∩ coNP
P

NP-completeness and $\mathrm{NP} \cap \mathrm{coNP}$

- Assume $\mathrm{NP} \neq \mathrm{coNP}$
- Under this assumption, we will prove that there are no NP-complete languages in $\mathrm{NP} \cap \mathrm{coNP}$
- This will provide evidence that FACTOR is not NP-complete


coNP-hard
NP-hard
coNP-complete
NP-complete
coNP
NP
FACTOR
NP ∩ coNP
P

coNP is closed under reductions

- Let $Y_1, Y_2 \subseteq \{0,1\}^*$

Lemma: If $Y_1 \leq_{\mathrm{P}} Y_2$ and $Y_2 \in \mathrm{coNP}$, then $Y_1 \in \mathrm{coNP}$

- **Proof:** Since $Y_2 \in \mathrm{coNP}$, there is a polynomial-time "co-nondeterministic" Turing machine M that decides Y_2
- Given $w \in \{0,1\}^*$, compute $w' = \Psi(w)$, then run M on w'

NP-completeness and $\mathbf{NP \cap coNP}$

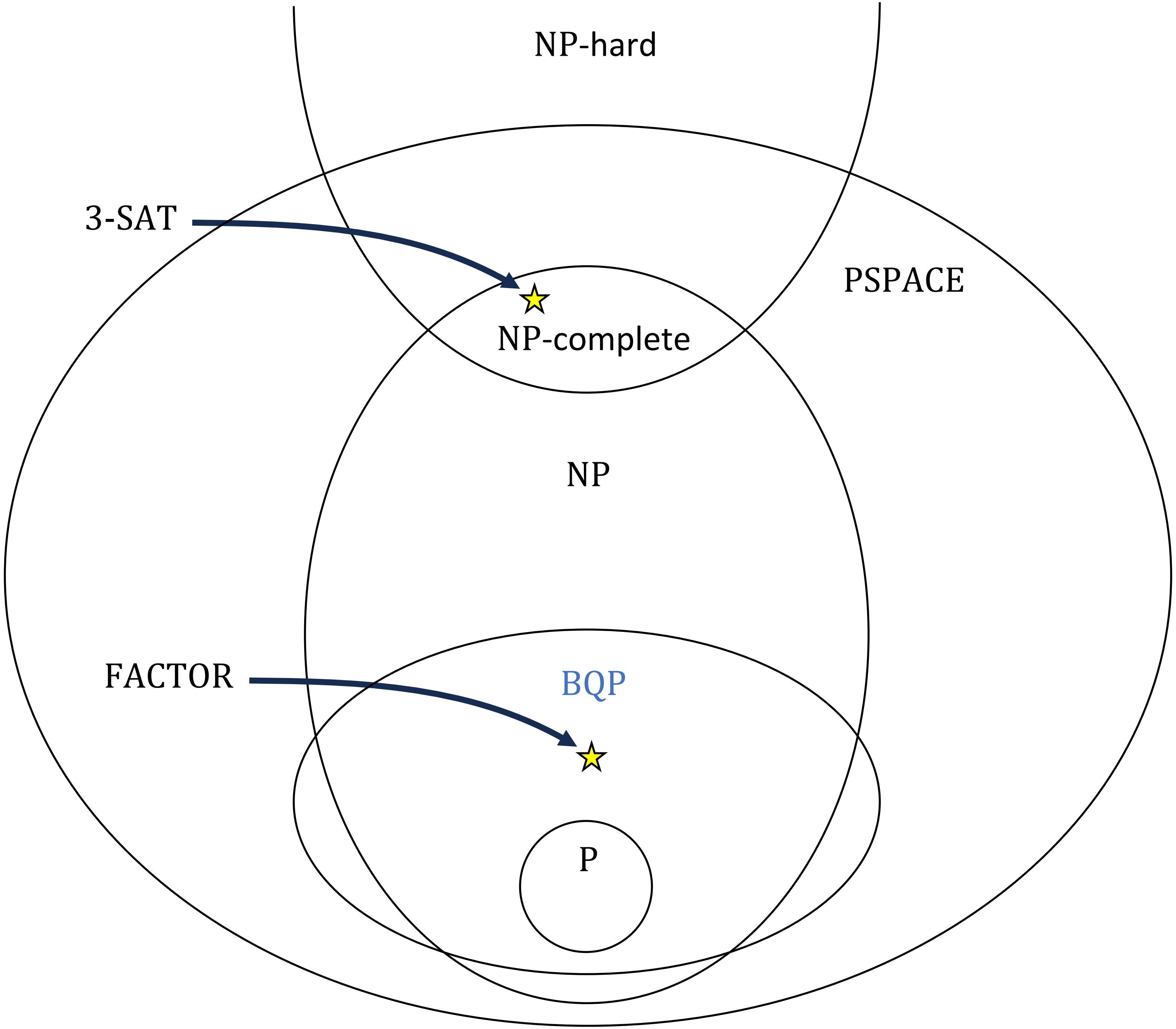
- Let $Y \in \mathrm{NP} \cap \mathrm{coNP}$

Claim: If Y is NP-complete, then $\mathrm{NP} = \mathrm{coNP}$

- **Proof:** For any $Z \in \mathrm{NP}$, we have $Z \leq_{\mathrm{P}} Y$ and $Y \in \mathrm{coNP}$
- By the lemma, $Z \in \mathrm{coNP}$, so $\mathrm{NP} \subseteq \mathrm{coNP}$
- By symmetry, we also have $\mathrm{coNP} \subseteq \mathrm{NP}$

Quantum computing is not a panacea

- $\mathrm{FACTOR} \in \mathrm{BQP}$, but FACTOR is probably not NP-complete
- In fact, it is conjectured that $\mathrm{NP} \not\subseteq \mathrm{BQP}$
- In this case, even a fully-functional quantum computer would not be able to solve NP-complete problems in polynomial time
- Even quantum computers have limitations


NP-hard
3-SAT
PSPACE
NP-complete
NP
FACTOR
BQP
P

Limitations of quantum computers

- We have developed several techniques for identifying hardness
    - Undecidability
    - EXP-completeness
    - NP-completeness
- Those techniques are all still applicable even in a world with fully-functional quantum computers!
- Complexity theory is intended to be "future-proof" / "timeless"

Which problems

can be solved

through computation?

~~CLASSICAL~~

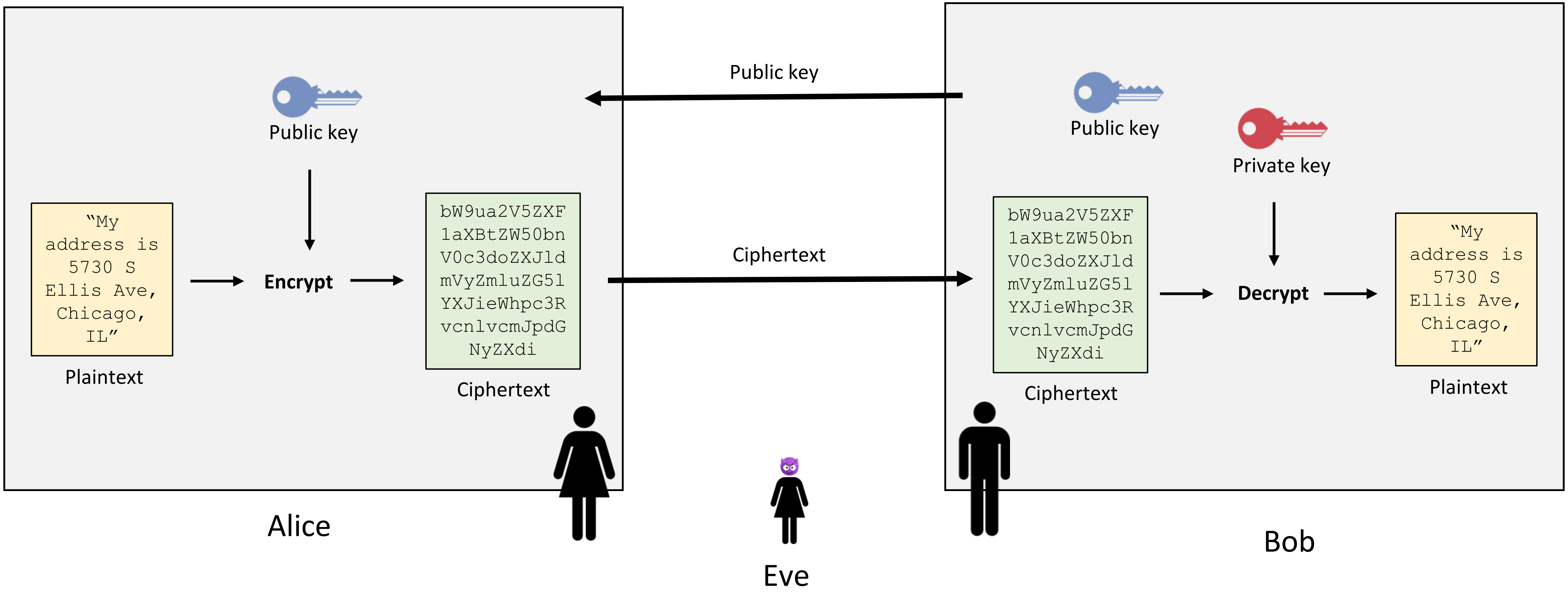
Intractability

- **Main topic of this course:** How to identify intractability
- **Previous few days:** How to cope with intractability
- **Up next:** How to exploit intractability

Cryptography

Public-key encryption

- How can Alice send a private message to Bob?



“encrypt”

“keys” “decrypt”

Public-key encryption scheme

- **Definition:** A simplified public-key encryption scheme is a triple (K, E, D), where:
  - $K \subseteq \{0,1\}^* \times \{0,1\}^*$ and $E, D: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$
  - For every $w \in \{0,1\}^*$ and every $(k_{\text{pub}}, k_{\text{priv}}) \in K$, we have $D\left(k_{\text{priv}}, E(k_{\text{pub}}, w)\right) = w$
  - E and D can be computed in polynomial time
  - For every $(k_{\text{pub}}, k_{\text{priv}}) \in K$, we have $|k_{\text{pub}}| = |k_{\text{priv}}|$
    - Intuition: Bigger keys ⇒ better security but slower encryption / decryption

Decrypting without k_{priv}

- Let (K, E, D) be a simplified public-key encryption scheme
- **Claim:** There exists $D_{\text{Eve}}: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that for every $w \in \{0,1\}^*$ and every $(k_{\text{pub}}, k_{\text{priv}}) \in K$, we have

$$D_{\text{Eve}}\left(k_{\text{pub}}, E(k_{\text{pub}}, w)\right) = w$$

- **Proof:** If $E(k_{\text{pub}}, w) = E(k_{\text{pub}}, w') = y$, then $w = D(k_{\text{priv}}, y) = w'$ ✔

Complexity theory to the rescue? ⌛

- Decrypting without k_{priv} is always possible 🙁
- 1970s discovery: There are public-key encryption schemes such that decrypting without k_{priv} seems to be intractable! 🙂
  - E.g., “RSA”
- Foundational technology for internet age
- Can we prove that these public-key encryption schemes are secure?

Cryptography and P vs. NP

- Let (K, E, D) be a simplified public-key encryption scheme
- There is a function D_{Eve} such that $D_{\mathrm{Eve}}\left(k_{\mathrm{pub}}, E(k_{\mathrm{pub}}, w)\right) = w$

Theorem: If $\mathrm{P} = \mathrm{NP}$, then D_{Eve} can be computed in polynomial time 🙁

Cryptography and P vs. NP

Theorem: If $\mathrm{P} = \mathrm{NP}$, then D_{Eve} can be computed in polynomial time 😦

- **Proof:** Let $Y = \{\langle k_{\mathrm{pub}}, y, w\rangle : \text{there exists } z \text{ such that } E(k_{\mathrm{pub}}, wz) = y\}$
- $Y \in \mathrm{NP}$: the plaintext is the certificate
- We are assuming $\mathrm{P} = \mathrm{NP}$, so therefore $Y \in \mathrm{P}$
- Therefore, Eve can construct the plaintext bit-by-bit in polynomial time

Cryptography and P vs. NP

- Disclaimer: The preceding discussion of public-key encryption is simplified
  - E.g., where do the keys come from?
- Nevertheless, the main message is accurate:
- If $\mathrm{P} = \mathrm{NP}$, then secure public-key encryption is impossible!

Cryptography and P vs. NP

- Almost all theoretical cryptography assumes $\mathrm{P} \neq \mathrm{NP}$ and more!
- This might make you feel concerned about the uncertain foundations of computer security…
- Or, it might make you feel more confident that $\mathrm{P} \neq \mathrm{NP}$, considering how hard people try to break cryptosystems